rocket gang

Traditional Project Management Meets Agile: Can't We All Get Along

Dr. Harry Koehnemann
Director of Technology
Rocket Gang
harry@rocketgang.com

Mark Coats
Chief Software Engineer
General Dynamics C4S
Mark.Coats@gdc4s.com

1

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **MAY 2011**	2. REPORT TYPE		3. DATES COVERED **00-00-2011 to 00-00-2011**
4. TITLE AND SUBTITLE **Traditional Project Management Meets Agile: Can't We All Get Along**		5a. CONTRACT NUMBER	
		5b. GRANT NUMBER	
		5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)		5d. PROJECT NUMBER	
		5e. TASK NUMBER	
		5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Rocket Gang,14362 N. Frank Lloyd Wright Blvd,Scottsdale,AZ,85260**		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)	
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**			
13. SUPPLEMENTARY NOTES **Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License**			
14. ABSTRACT			
15. SUBJECT TERMS			

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

Agenda

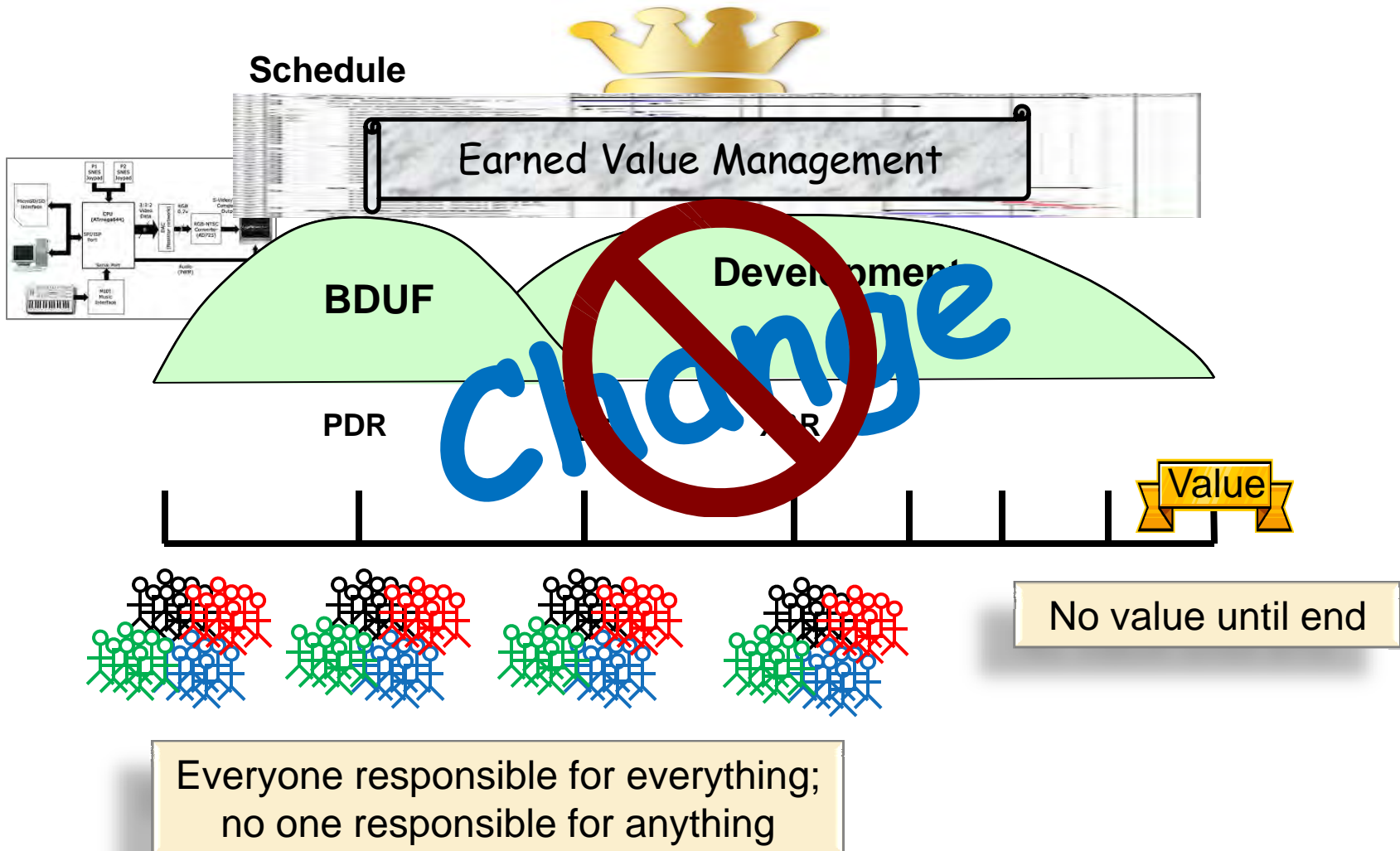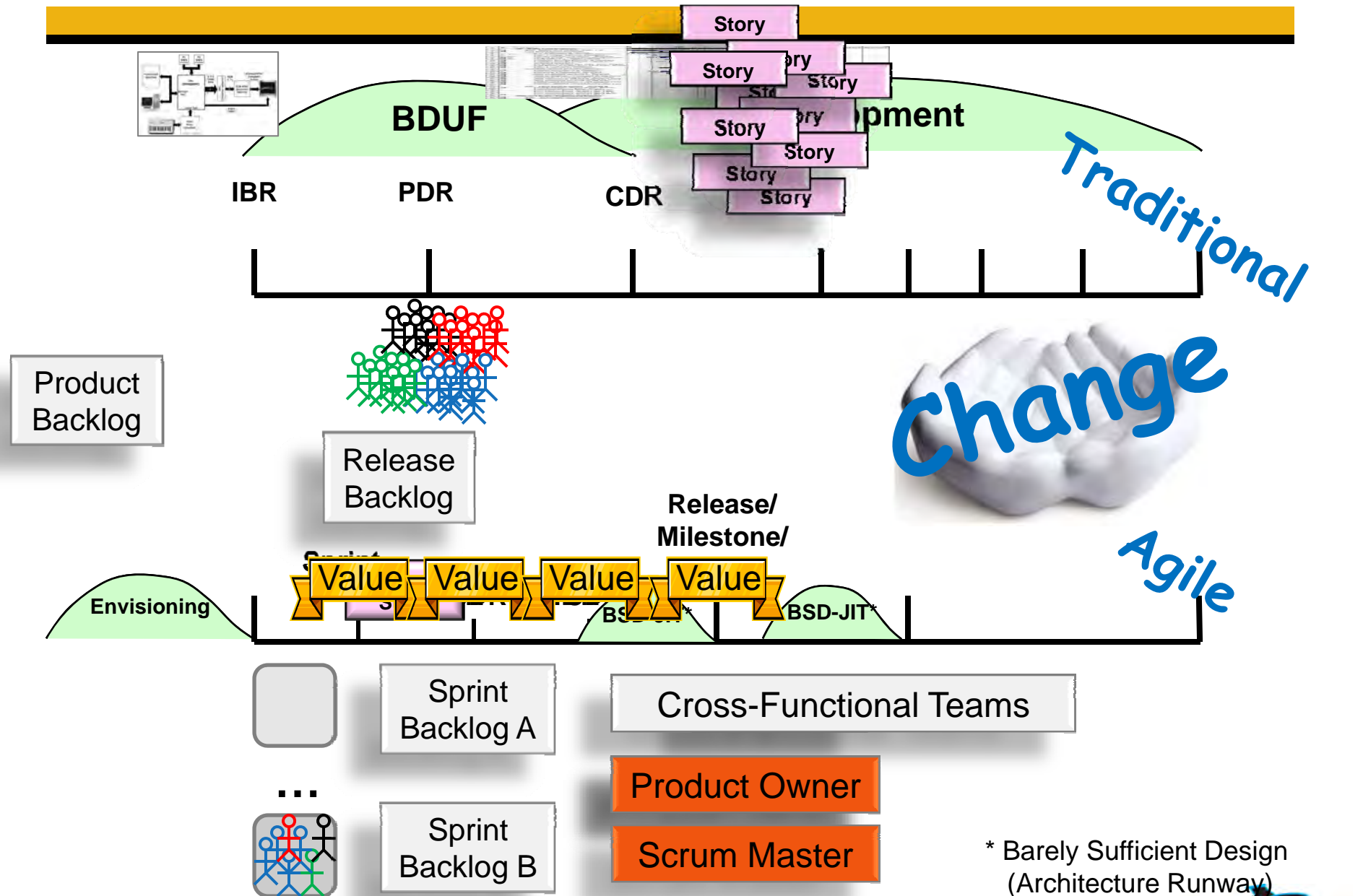- Traditional planning and it's challenges

    ↘ Focus on team organization, planning, requirements, and architecture

- Making traditional planning agile

- Transitioning from traditional to agile

Traditional Development

Schedule

Earned Value Management

BDUF

Development

PDR

Change

Value

No value until end

Everyone responsible for everything;
no one responsible for anything

3

rocket gang

Agile Development

BDUF

IBR   PDR   CDR

Traditional

Change

Product
Backlog

Release
Backlog

Release/
Milestone/

Agile

Envisioning

Value   Value   Value   Value

BSD-JIT*   BSD-JIT*

Sprint
Backlog A

Cross-Functional Teams

...

Sprint
Backlog B

Product Owner

Scrum Master

* Barely Sufficient Design
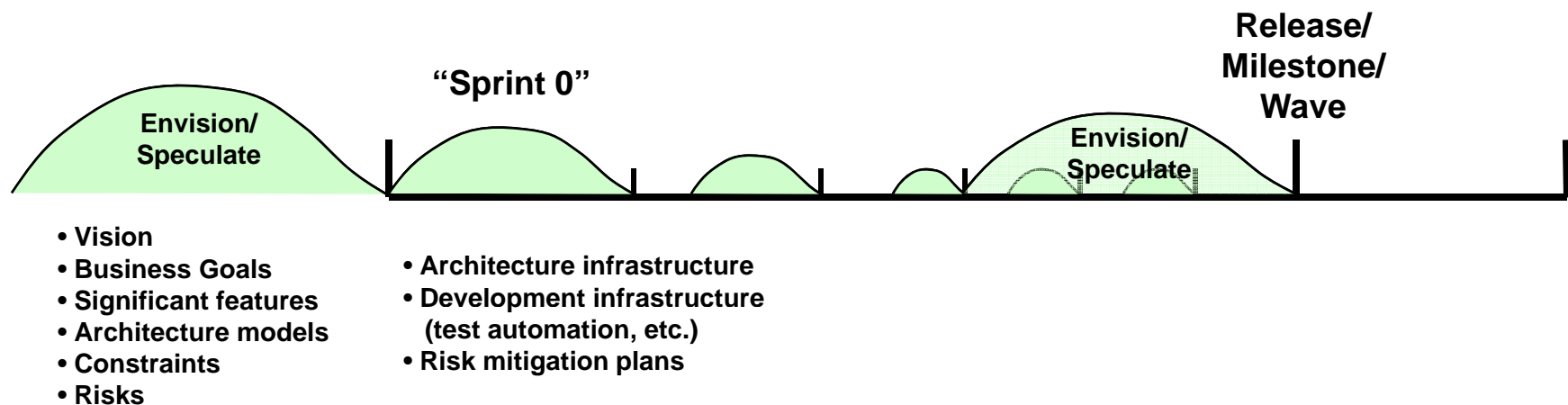(Architecture Runway)

4

rocket gang

Agile Value Points

- Defines success as <u>delivering business value</u>, vs. meeting schedule

- Continuous, <u>frequent collaboration</u> points between dev and business

- Encourages frequent <u>feedback and learning</u>

  ↘ Allows projects to fail fast

  ↘ Quickly adapts to changes

- Accommodates uncertainty, risk, change

- Plans work and organizes teams around system behavior

- Details requirements, design, schedule at right time
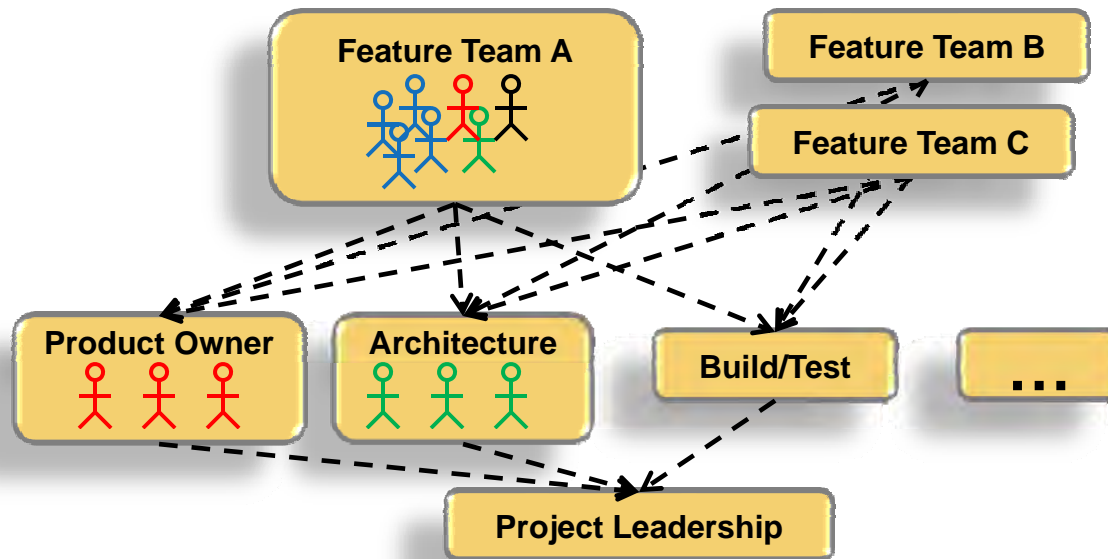
rocket gang

Transitioning to Agile

- Addresses team organization, planning, requirements, architecture

  - Continually define barely-sufficient requirements and architecture runway for feature teams

- Envision/Speculate (vs. Initiate/Plan)

  ↘ Brainstorm on product vision and promising implementation approach

- Explore (vs. Execute)

  ↘ Use iterations to explore solution and adapt to new discoveries

  ↘ Vs. preserve original baselines at all costs

Envision/ Speculate

"Sprint 0"

Release/ Milestone/ Wave

Envision/ Speculate

- **Vision**
- **Business Goals**
- **Significant features**
- **Architecture models**
- **Constraints**
- **Risks**

- **Architecture infrastructure**
- **Development infrastructure (test automation, etc.)**
- **Risk mitigation plans**

6

rocket gang

Transitioning to Agile Team Organization

- Organize around behavior, not structure or domain

- Teams self-manage

  ↘ Commit to features, decompose work, assign tasks

  ↘ Desire to keep co-located!!

- Strive to push decisions out to feature teams

  ↘ Specialty teams responsible for broader system issues and decisions

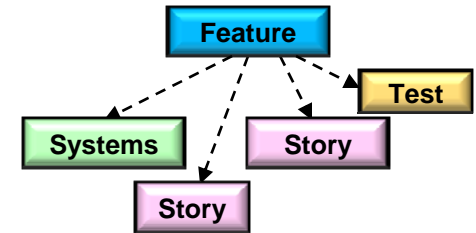- Coupled features may require high collaboration between teams

Feature Team A

Feature Team B

Feature Team C

Feature Teams

Product Owner

Architecture

Build/Test

...

"Specialty Teams", possibly ad-hoc
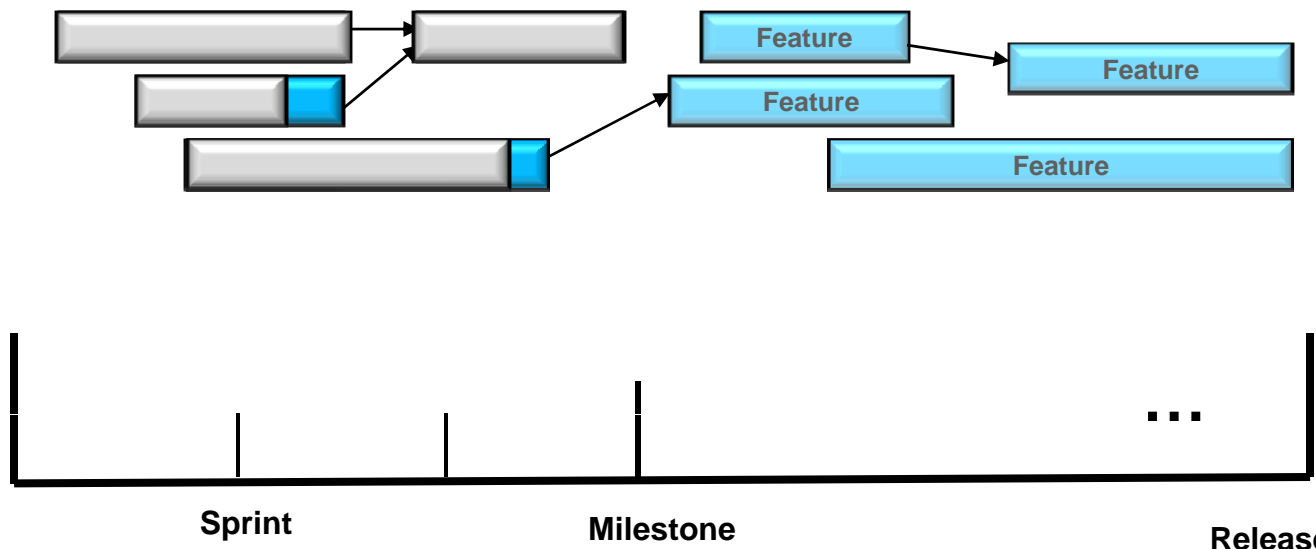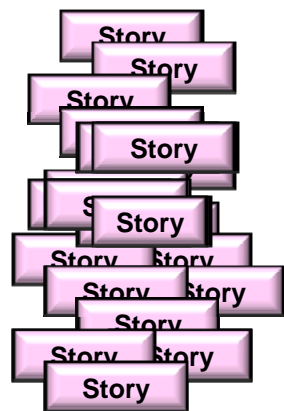
Project Leadership

rocket gang

Transition to Agile Planning

- Plans (WBS) define behavior
  - ↘ Releases plan features (20-100 day effort)
  - ↘ Decompose into stories (2-10 day)
- Items become more detailed closer to scheduling
- Use milestones/waves as synchronization points
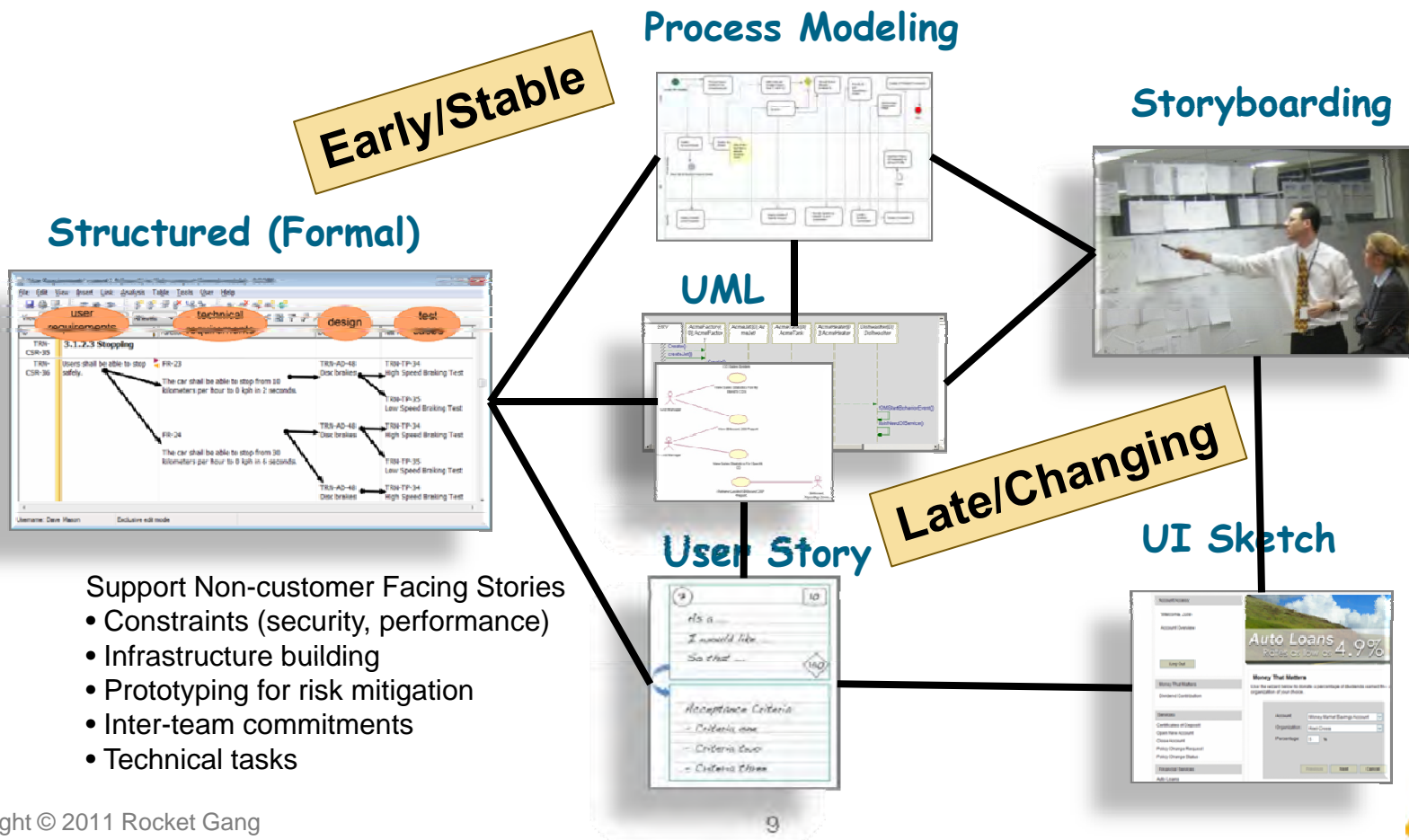 - ↘ Each iteration we learn more about team's ability to meet schedule

Transitioning to Agile Requirements

- Envisioning defines constraints, significant features

- Exploring plans and details requirements at right time

- Need additional story types to support large, complex systems

Early/Stable

Process Modeling

Storyboarding

Structured (Formal)

UML

Late/Changing

User Story

UI Sketch

Support Non-customer Facing Stories
- Constraints (security, performance)
- Infrastructure building
- Prototyping for risk mitigation
- Inter-team commitments
- Technical tasks

rocket gang

Transitioning to Agile Architecture

- Complex systems also develop their infrastructure

  ↘ Responsible for significant functionality *below the waterline*

- Prioritize stories that extend and validate infrastructure

  ↘ Includes rich set of <u>automated tests</u>

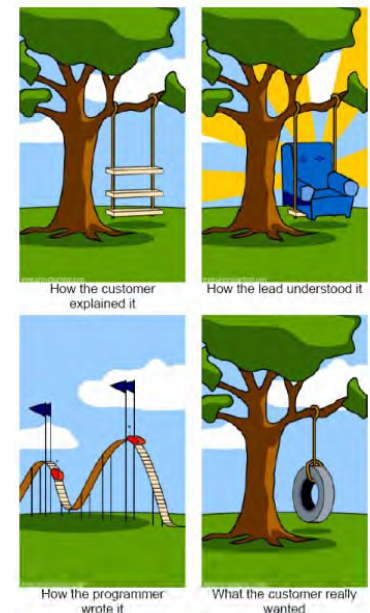 ↘ Similar in spirit to the "Sprint 0" agile practice ("Milestone 0")
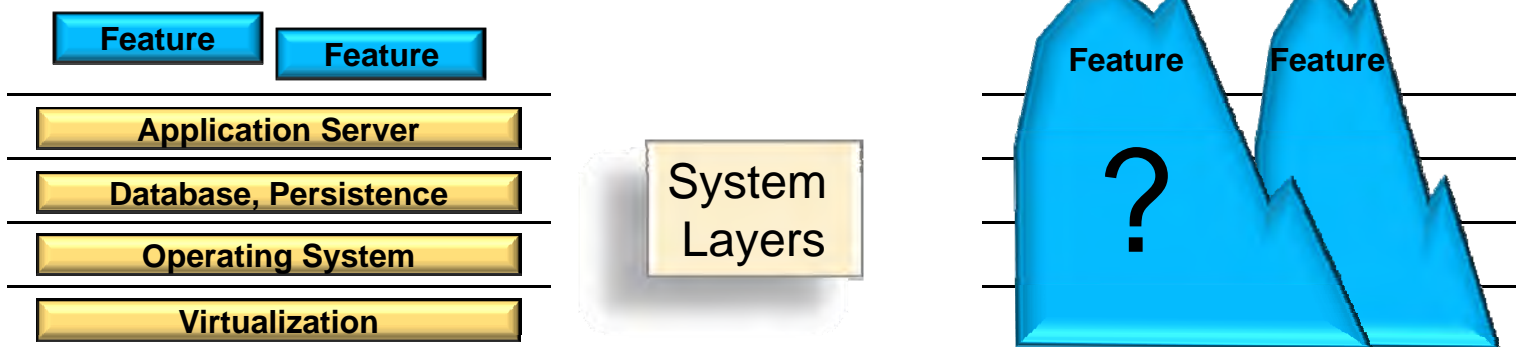
Caution on Organizational Change

- Great process cannot make up for having the wrong people

- Need capable people with self-discipline

  ↘ Self-motivated, passionate about solving problems

  ↘ Multi-skilled, adaptive, continuous improvers

  ↘ Proactive, sense of ownership

  ↘ Communicate, collaborate

- Coaching/modeling is key to organizational change

"Most organizations build their bureaucratic rules to manage the few wrong people on the bus, which drives away the right people on the bus, which in turn increases the need for more bureaucratic rules, and so on" -- Jim Collins (2001)

rocket gang

Recommendations

- Blend strengths
  - ↘ Agile: continuous feedback, learning, adaptive planning, behavior focus
  - ↘ Traditional: up-front assumptions

- Consider agility principles
  - ↘ Knowledge and running system are the goals, not documents and schedules
  - ↘ Focus on feedback and learning – problem, solution, development process
  - ↘ Use continuous, inspect-adapt cycles
  - ↘ Plan, track, execute small work
  - ↘ Organize teams around features
  - ↘ Lead teams, let teams manage themselves

- Repeatable wants to schedule, reliable wants to deliver value
  - ↘ Focus more on execution, less on management/control

How the customer explained it

How the lead understood it

How the programmer wrote it

What the customer really wanted

rocket gang

Conclusions

- Fill the gap between project plans and developer tasks

  ↘ Ensure plans are based on reality

  ↘ Automate progress from development work to project-level plans

- Agile changes how we:

  ↘ Plan   Behavior-focused vs. structure/document-focused

  ↘ Measure   Business value vs. percent complete

  ↘ Manage   Self-managed teams vs. Project Manager

  ↘ Track   Feature-centric vs. plan-centric

  ↘ Build & Test Continuously vs. at end

  ↘ Define success Make business smile (value ☺) vs. execute the plan

Thank You

Dr. Harry Koehnemann
Director of Technology
Rocket Gang
harry@rocketgang.com

Mark Coats
Chief Software Engineer
General Dynamics C4S
Mark.Coats@gdc4s.com

rocket gang